

### **Amendments to the Specification:**

Please replace paragraph [0032] with the following marked up paragraph:

[0032] In an embodiment of the invention, a dynamic binary translator 203 may be used to ensure the program portability to a host platform 210. Similar to the configuration described in FIG. 1, a platform-independent code 201 may encapsulate a foreign thread code 202. The foreign thread code 202 is translated by the dynamic binary translator 203 to a host native code 204. The host native code 204 may then be executed by the host platform 210 and supported by the underlying components such as other libraries 211, the thread libraries 212, an operating system 213, a host ISA 214, and a host processor 215. Here, the ~~read thread~~ thread libraries 212 may be a foreign thread library capable of managing and supporting the foreign thread code 202.

Please replace paragraph [0044] with the following marked up paragraph:

[0044] To associate the IA-32 thread and the IPF thread, the program calls `mie_complete_thread_init(PARENT)` 406. In this function, the parent IPF thread completes its thread initialization by creating the IA-32 thread beginning from the clone point. Furthermore, the child IPF thread also complete its thread initialization by calling ~~`mie_complete_thread`~~ `mie_complete_thread init(CHILD)` 406. In this function, the IA-32 thread is translated and the resource context also begins from the clone point until the execution returns at the middle tier layer 410.